

# Lec1 词嵌入与早期语言模型

- 从数值映射到词向量编码
- N-grams语言模型
- RNN语言模型
  - 模型架构及数学原理
  - 代码实现
  - 对训练隐患（梯度问题）的数学解释
- LSTM语言模型
  - 模型架构
  - 代码实现

# Lec2 早期Transformer架构

- 基于RNN的Seq2Seq框架
- 基于Transformer的Seq2Seq
  - 注意力机制及Causal掩码
  - 位置编码（早期的正弦曲线编码）
  - Feed-Forward Network
  - 残差连接
  - 代码实现
  - 复杂度分析

# Lec3 基于Transformer的现代模型架构及其改进

- 基于Transformer的新模型架构
  - GPT
    - Decoder模型架构
    - 运行方式及训练方法
  - BERT
    - Encoder模型架构
    - 运行方式及训练方法

- 模块改进
  - 归一化策略：Pre-Norm, RMSNorm
  - 位置编码改进：RoPE
  - FFN改进：SwiGLU
  - Attention改进：KV Cache、Sparse Attention、Linear Attention、Flash Attention
- 超参选择
  - $d_{ff}$ 与 $d_{model}$ 比值
  - $num\_heads * head\_dim$ 与 $d_{model}$ 比值
  - $d_{model}$ 与 $n_{layer}$ 比值

## Lec4 BPE分词

总体按照

[https://github.com/kuangpenghao/ADG\\_Build\\_Language\\_Model\\_Lectures/blob/main/Lec1\\_BPE.pdf](https://github.com/kuangpenghao/ADG_Build_Language_Model_Lectures/blob/main/Lec1_BPE.pdf) 内容，不做明显改动

- 原始分词算法的问题
- BPE算法原理
- 时间复杂度优化
- 代码讲解与运行示例

## Lec5 架构实现

总体按照

[https://github.com/kuangpenghao/ADG\\_Build\\_Language\\_Model\\_Lectures/blob/main/Lec3\\_Build\\_Transformer.pdf](https://github.com/kuangpenghao/ADG_Build_Language_Model_Lectures/blob/main/Lec3_Build_Transformer.pdf) 内容，不做明显改动

讲解了Transformer Decoder Model的完整代码实现

## Lec6 模型训练

- 资源计算
  - 峰值内存与batch size
  - 计算量（FLOPs）估计
  - 训练耗时估计

剩余内容总体按照

[https://github.com/kuangpenghao/ADG\\_Build\\_Language\\_Model\\_Lectures/blob/main/Lec4\\_Train\\_and\\_Decode.pdf](https://github.com/kuangpenghao/ADG_Build_Language_Model_Lectures/blob/main/Lec4_Train_and_Decode.pdf) 内容，不做明显改动

# Lec7 Hugging Face

利用Hugging Face框架实现模型的自定义修改与训练

- Hugging Face介绍
- 模型训练的原理介绍
  - 原始模型代码
  - 自定义模型的注册过程
  - 模型训练脚本代码
- 自定义模型任务
  - 自定义模型的实现
  - 模型训练

# Lec8 wandb调参

利用wandb进行模型超参数调优

- wandb介绍
- 调参的原理介绍与操作
  - yaml脚本的编写
  - 超参生成的原理
  - 训练脚本的改动（wandb库的使用）
  - 可视化调参

# Project 1

计划分成1.1、1.2、1.3共三次。

## 1.1 BPE

- 内容：实现完整的BPE分词算法及算法优化。题面可搬运CS336的相关部分
- 测试：可搬运CS336的本地测试

- 提交与评分：考虑在gradescope设置代码提交通道，仅作为提交用；具体的评分（如检查测试点正确率、代码质量、相关实现细节思路的检查）由线下check完成

## 1.2 Transformer

- 内容：实现完整的Transformer Decoder Model模型架构与模型训练解码脚本。题面可搬运CS336的相关部分
- 测试：可搬运CS336的本地测试
- 提交与评分：考虑在gradescope设置代码提交通道，仅作为提交用；具体的评分（如检查测试点正确率、代码质量、相关实现细节思路的检查）由线下check完成

## 1.3 Hugging Face

- 内容：给定一个简单的Transformer变体模型修改需求（如：修改Attention模块的实现或某种残差连接机制），利用Hugging Face框架实现变体模型的修改与训练；利用wandb对给定的超参数进行调优。
- 评分标准（线下check）
  - 本次作业评分应较为宽松，分值占比较小，仅仅作为一次简单实操，能完整实现即可给分，仅对明显的实现问题（如实现思路明显有误或未完成）进行扣分。
  - Part1：变体模型实现思路正确，可正常训练即可给分
  - Part2：能正常进行调参，展示出可视化调参界面即可给分